


[> home](#) [> about](#) [> feedback](#) [> login](#)

US Patent & Trademark Office



Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: [virtual AND address AND memory AND map]

Found 2,746 of 114,152 searched.

Warning: Maximum result set of 200 exceeded. Consider refining.

Search within Results

 [> Advanced Search](#)
[> Search Help/Tips](#)
Sort by: [Title](#) [Publication](#) [Publication Date](#) [Score](#)Results 1 - 20 of 200 [short listing](#)
[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)


- 1** [The integration of virtual memory management and interprocess communication in Accent](#) 100%

Robert Fitzgerald , Richard F. Rashid
ACM Transactions on Computer Systems (TOCS) May 1986
 Volume 4 Issue 2
 The integration of virtual memory management and interprocess communication in the Accent network operating system kernel is examined. The design and implementation of the Accent memory management system is discussed and its performance, both on a series of message-oriented benchmarks and in normal operation, is analyzed in detail.
- 2** [Machine-independent virtual memory management for paged uniprocessor and multiprocessor architectures](#) 100%


Richard Rashid , Avadis Tevanian , Michael Young , David Golub , Robert Baron
Proceedings of the second international conference on Architectural support for programming languages and operating systems October 1987
 Volume 15 , 22 , 21 Issue 5 , 10 , 4
- 3** [Hardware architecture for recursive Virtual Machines](#) 99%

Gerald Belpaire , Nai-Ting Hsu
Proceedings of the 1975 annual conference January 1975
 In order to support Virtual Machine (VM) Systems on most current computer systems, the Virtual Machine Monitor (VMM) is the only process allowed to reference directly a set of registers, called resource management registers (RMR). Any access operations to those registers executed by any other processes must be detected and executed interpretively by the VMM. This way of implementing VMs is imposed by inadequate hardware mechanism for

protection. In this paper, this problem is studied and a ...

4 The Multics virtual memory: concepts and design

98%

 A. Bensoussan , C. T. Clingen , R. C. Daley


Communications of the ACM May 1972

Volume 15 Issue 5

As experience with use of on-line operating systems has grown, the need to share information among system users has become increasingly apparent. Many contemporary systems permit some degree of sharing. Usually, sharing is accomplished by allowing several users to share data via input and output of information stored in files kept in secondary storage. Through the use of segmentation, however, Multics provides direct hardware addressing by user and system programs of all information, indepe ...

5 The duality of memory and communication in the implementation of a multiprocessor operating

98%

 system

M. Young , A. Tevanian , R. Rashid , D. Golub , J. Eppinger

ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles November 1987

Volume 21 Issue 5

Mach is a multiprocessor operating system being implemented at Carnegie-Mellon University. An important component of the Mach design is the use of memory objects which can be managed either by the kernel or by user programs through a message interface. This feature allows applications such as transaction management systems to participate in decisions regarding secondary storage management and page replacement. This paper explores the goals, design and implementation of Mach and it ...

6 Third Generation Computer Systems

98%

 Peter J. Denning


ACM Computing Surveys (CSUR) December 1971

Volume 3 Issue 4

The common features of third generation operating systems are surveyed from a general view, with emphasis on the common abstractions that constitute at least the basis for a "theory" of operating systems. Properties of specific systems are not discussed except where examples are useful. The technical aspects of issues and concepts are stressed, the nontechnical aspects mentioned only briefly. A perfunctory knowledge of third generation systems is presumed.

7 Kernels: Practical, transparent operating system support for superpages

97%

 Juan Navarro , Sitaram Iyer , Peter Druschel , Alan Cox

ACM SIGOPS Operating Systems Review December 2002

Volume 36 Issue SI


Most general-purpose processors provide support for memory pages of large sizes, called superpages. Superpages enable each entry in the translation lookaside buffer (TLB) to map a large physical memory region into a virtual address space. This dramatically increases TLB coverage, reduces TLB misses, and promises performance improvements for many applications. However, supporting superpages poses several challenges to the operating system, in terms of superpage allocation and promotion tradeoffs, ...

- 8 A formal description of the UNIX operating system 97%
Thomas W. Doeppner , Alessandro Glacalone
Proceedings of the second annual ACM symposium on Principles of distributed computing August 1983
In this paper we discuss our approach to a formal description of the UNIX operating system [Rit78a] [Rit78b] [Tho78], using Milner's Calculus of Communicating Systems (CCS) [Mil80]. The paper focuses on the problems one encounters and the decisions one has to make when describing a system such as UNIX. We believe that the problems that arise in defining such a system are much less well understood than those, for example, related to the formalization of programmin ...
- 9 The M-Machine multicomputer 97%
Marco Fillo , Stephen W. Keckler , William J. Dally , Nicholas P. Carter , Andrew Chang , Yevgeny Gurevich , Whay S. Lee
Proceedings of the 28th annual international symposium on Microarchitecture December 1995
- 10 The sawmill framework for virtual memory diversity 96%
Mohit Aron , Jochen Liedtke , Kevin Elphinstone , Yoonho Park , Trent Jaeger , Luke Deller
Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture January 2001
Volume 23 Issue 4
We present a framework that allows applications to build and customize VM services on the L4 microkernel. While the L4 microkernel's abstractions are quite powerful, using these abstractions effectively requires higher-level paradigms. We propose the dataspace paradigm which provides a modular VM framework. The modularity introduced by the dataspace paradigm facilitates implementation and permits dynamic configurability. Initial performance results from a prototype are promising.
- 11 File servers for network-based distributed systems 96%
Liba Svobodova
ACM Computing Surveys (CSUR) December 1984
Volume 16 Issue 4
- 12 From RIG to Accent to Mach: the evolution of a network operating system 95%
Richard F. Rashid
Proceedings of 1986 fall joint computer conference on Fall joint computer conference November 1999
- 13 Fast detection of communication patterns in distributed executions 95%
Thomas Kunz , Michiel F. H. Seuren
Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research November 1997
Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated

occurrences of non-trivial commun ...

14 A structured specification of a hierarchical operating system

95%

 Ashok R. Saxena , Thomas H. Bredt


ACM SIGPLAN Notices , Proceedings of the international conference on Reliable software April 1975

Volume 10 Issue 6

This paper applies the concepts of hierarchical levels of abstraction and structured programming to the design of a large program system. An operating system for a multi-processor installation is specified that supports a large number of concurrently active processes and provides a virtual store for them. The specification is in an extended version of PASCAL, a high-level language.

15 Architecture of virtual machines

95%


 R. P. Goldberg

Proceedings of the workshop on virtual computer systems March 1973

In this paper we develop a model which represents the addressing of resources by processes executing on a virtual machine. The model distinguishes two maps: the \emptyset -map which represents the map visible to the operating system software running on the virtual machine, and the f-map which is invisible to that software but which is manipulated by the virtual machine monitor running on the real machine. The \emptyset -map maps process names into resource names and the f-map maps virtual resou ...

16 The file system of an integrated local network

95%


 Paul J. Leach , Paul H. Levine , James A. Hamilton , Bernard L. Stumpf

Proceedings of the 1985 ACM thirteenth annual conference on Computer Science March 1985

The distributed file system component of the DOMAIN system is described. The DOMAIN system is an architecture for networks of personal workstations and servers which creates an integrated distributed computing environment. The distinctive features of the file system include: objects addressed by unique identifiers (UIDs); transparent access to objects, regardless of their location in the network; the abstraction of a single level store for accessing all objects; and the layering of a networ ...

17 Distributed operating systems

94%

 Andrew S. Tanenbaum , Robbert Van Renesse

ACM Computing Surveys (CSUR) December 1985

Volume 17 Issue 4

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

18 Garbage collection for virtual memory computer systems

94%

 H. D. Baecker

Communications of the ACM November 1972

Volume 15 Issue 11

In list processing there is typically a growing demand for space during program execution. This paper examines the practical implications of this growth within a virtual memory computer system, proposes two new garbage collection techniques for virtual memory systems, and compares them with traditional methods by discussion and by simulation.

19 Formal requirements for virtualizable third generation architectures

94%

 Gerald J. Popek , Robert P. Goldberg**Communications of the ACM** July 1974

Volume 17 Issue 7

Virtual machine systems have been implemented on a limited number of third generation computer systems, e.g. CP-67 on the IBM 360/67. From previous empirical studies, it is known that certain third generation computer systems, e.g. the DEC PDP-10, cannot support a virtual machine system. In this paper, model of a third-generation-like computer system is developed. Formal techniques are used to derive precise sufficient conditions to test whether such an architecture can support virtual mach ...

20 Avoiding conflict misses dynamically in large direct-mapped caches

94%

 Brian N. Bershad , Dennis Lee , Theodore H. Romer , J. Bradley Chen**Proceedings of the sixth international conference on Architectural support for programming languages and operating systems** November 1994

Volume 29 , 28 Issue 11 , 5

This paper describes a method for improving the performance of a large direct-mapped cache by reducing the number of conflict misses. Our solution consists of two components: an inexpensive hardware device called a Cache Miss Lookaside (CML) buffer that detects conflicts by recording and summarizing a history of cache misses, and a software policy within the operating system's virtual memory system that removes conflicts by dynamically remapping pages whenever large numbers of conflict miss ...

Results 1 - 20 of 200

[short listing](#)
Prev
Page[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)[10](#)
Next
Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[> home](#) [> about](#) [> feedback](#) [> login](#)

US Patent & Trademark Office



Try the *new* Portal design
Give us your opinion after using it.

Search Results


Search Results for: **[global breakpoint]**

Found 8 of 114,152 searched.

Search within Results

  [> Advanced Search](#)[> Search Help/Tips](#)Sort by: [Title](#) [Publication](#) [Publication Date](#) [Score](#)  [Binder](#)Results 1 - 8 of 8 [short listing](#)1 [An interactive debugging facility for GPSS](#)


87%

 James O. Henriksen**Proceedings of the 9th conference on Winter simulation - Volume 1** December 1977

This paper describes the interactive debugging feature of GPSS/H (1), a new implementation of GPSS for IBM 360/370 computers. In GPSS/H, interactive debugging of a simulation model is carried out by means of a simple, but powerful command language. Commands are provided to selectively display model data on a terminal, to set breakpoints at arbitrary points in a model, and to step through a model one or more blocks at a time. An overview of the command language and an illustrative example of ...

2 [Fast detection of communication patterns in distributed executions](#)


77%

 Thomas Kunz , Michiel F. H. Seuren**Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research** November 1997


Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

3 [An algebraic specification of the partial orders generated by concurrent Ada computations](#)

77%

 D. Bryan**Proceedings of the conference on Tri-Ada '89: Ada technology in context: application, development, and deployment** January 19894 [Debugging concurrent programs](#)

77%

 Charles E. McDowell , David P. Helmbold

ACM Computing Surveys (CSUR) December 1989

Volume 21 Issue 4

The main problems associated with debugging concurrent programs are increased complexity, the "probe effect," nonrepeatability, and the lack of a synchronized global clock. The probe effect refers to the fact that any attempt to observe the behavior of a distributed system may change the behavior of that system. For some parallel programs, different executions with the same data will result in different results even without any attempt to observe the behavior. Even when the behavior can be ...

5 The Mantis parallel debugger


77%

 Steven S. Lumetta , David E. Culler

Proceedings of the SIGMETRICS symposium on Parallel and distributed tools January 1996

6 Event and state-based debugging in TAU: a prototype


77%

 Sameer Shende , Janice Cuny , Lars Hansen , Joydip Kundu , Stephen McLaughry , Odile Wolf

Proceedings of the SIGMETRICS symposium on Parallel and distributed tools January 1996

7 ENF event predicate detection in distributed systems


77%

 Hsien-Kuang Chiou , Willard Korfhage

Proceedings of the thirteenth annual ACM symposium on Principles of distributed computing August 1994

8 An approach to reducing delays in recognizing distributed event occurrences

77%

 Madalene Spezialetti

ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging December 1991

Volume 26 Issue 12

Results 1 - 8 of 8 [short listing](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.